eOMP: Finding Sparser Representation by Recursively Orthonormalizing the Remaining Atoms

Yuanyi Xue, and Yao Wang

Abstract—Greedy algorithms for minimizing L0-norm of sparse decomposition have profound application impact on many signal processing problems. In the sparse coding setup, given the observations y and the redundant dictionary Φ, one would seek the most sparse coefficient (signal) x with a constraint on approximation fidelity. In this work, we propose a greedy algorithm based on the classic orthogonal matching pursuit (OMP) with improved sparsity on x and better recovery rate, which we name as eOMP. The key ingredient of the eOMP is recursively performing one-step orthonormalization on the remaining atoms, and evaluating correlations between residual and orthonormalized atoms. We show a proof that the proposed eOMP guarantees to maximize the residual reduction at each iteration. Through extensive simulations, we show the proposed algorithm has better exact recovery rate on i.i.d. Gaussian ensembles with Gaussian signals, and more importantly yields smaller L0-norm under the same approximation fidelity compared to the original OMP, for both synthetic and practical scenarios. The complexity analysis and real running time result also show a manageable complexity increase over the original OMP. We claim that the proposed algorithm has better practical perspective for finding more sparse representations than existing greedy algorithms.

I. INTRODUCTION

THE recent developments in sparse recovery have yielded profound impact on many old and new signal processing problems. In the sparse coding setting, people are looking for solving the classic inverse problem where the sampling matrix (or dictionary) Φ constructs an underdetermined system. Specifically, given $\Phi \in \mathbb{R}^{N \times M}$ where $M \gg N$ and observations $y \in \mathbb{R}^N$, it solves an $x \in \mathbb{R}^M$ via L0-norm minimization problem:

$$\begin{aligned} \min \quad & |x|_0 \\ \text{s.t.} \quad & \|y - \Phi x\|_2 < \epsilon \end{aligned} \tag{1}$$

The nature of non-convexity of the L0-norm leads to two genre of algorithms for solving the above problem. The first category of algorithms seeks to relax the non-convex L0-norm term into a more optimization friendly L1-norm, such as the well-known LASSO algorithm [1]. Although it has mathematical guarantee on global convergence, the L1-norm in fact does not precisely measure the sparsity, which is the number of non-zero coefficients in the solution vector. On the other hand, another group of scholars design a series of primarily greedy algorithms to minimize the L0-norm directly. An exemplar of such algorithms is the orthogonal matching pursuit [2], which iteratively finds the best correlated atom to the residual. More recently, several improved algorithms have been proposed. In [3] and [4], the authors proposed two similar algorithms (CoSaMP and subspace pursuit), respectively, that have been proven to have better exact recovery conditions and faster convergence, for signals with known sparsity.

Although the new algorithms are promising, we would like to note that they are mostly based on the following two assumptions: 1) all above cited work have proved exact recovery conditions that in general require the sampling matrix Φ to be nearly orthonormal (as measured by the restricted isometry property [5]); and 2) in the

Y. Xue, and Y. Wang are with the Department of Electrical and Computer Engineering, Polytechnic School of Engineering, New York University. E-mail: {yxue,yw523}@nyu.edu.

synthetic simulations demonstrated, the signals are often constructed with known sparsity. This is particularly true for [3] and [4], wherein the estimated true sparsity is needed as the algorithm input. We would like to note these assumptions hamper the usability of these algorithms in the practical sparse coding scenario, where the sampling matrix is usually highly correlated and the actual signal sparsity is unknown. Indeed, [3] and [4] are targeting a different version of Eq. 1, which tries to minimize the fidelity with an equality constraint on the L0-norm of x.

From this standpoint, we propose a greedy algorithm based on the classic OMP algorithm which better suits the scenario with unknown sparsity and correlated sampling matrix. The proposed algorithm recursively performs one-step orthonormalization on the remaining atoms with respect to previously chosen atoms in the dictionary at each iteration, so that the orthnormalized remaining atoms span over the null space of the current signal estimate. We show through extensive synthetic and real simulations that the proposed algorithm is able to achieve better sparsity compared to the classic OMP, with a manageable complexity increase. We name the proposed algorithm as eOMP because it uses embedded recursive orthonormalization.

The rest of the paper is organized as follows. In Sec. II, we lay out the details of the proposed eOMP algorithm and we show the proof that the proposed eOMP guarantees to maximize residual reduction at each iteration; in Sec. III, we provide extensive simulation results for synthetic simulations under two sampling matrix configurations, the i.i.d Gaussian random ensemble and the overcomplete DCT dictionary, as well as a practical simulation based on coding video blocks by finding a sparse representation over a highly redundant and correlated self-adaptive dictionary [6]; we show the complexity analysis and real running time statistics of the proposed algorithm in Sec. IV; we conclude the paper in Sec. V.

II. IMPROVING THE OMP ALGORITHM

In the classic OMP, at each iteration, the algorithm consists of the following three major steps: find the most correlated column (atom) among all remaining columns of the sampling matrix with the current residual and append the chosen atom index into the set of chosen atoms (known as the support), a least squares update of the coefficients corresponding to the new support, and at last the residual is updated using least squares coefficients. As all greedy algorithms, the OMP converges when the residual norm stops decreasing or reaches the preset ϵ shown in Eq. 1.

Before we jump to our proposed algorithm, we introduce an alternative way of implementing the original OMP algorithm by orthonormalizing the chosen atoms incrementally. Specifically, at each iteration, after selecting the chosen atom, we orthonormalize it with respect to the previous chosen atoms using the Gram-Schmidt algorithm. The coefficient with respect to this orthonormalized atom can be then obtained simply by the inner product (i.e. projection) of the current residual to this orthonormalized atom. This alternative way of implementing OMP is shown in Algorithm 1. We would like to note that although the solution vector corresponds to the orthonormalized atoms, finding the coefficients to the original chosen

atoms can be done after all atoms are chosen by a single least squares step.

Algorithm 1: Alternative OMP algorithm with incremental orthonormalization of chosen atoms

Data: $\Phi \in \mathbb{R}^{N \times M}, y \in \mathbb{R}^{N \times 1}, \epsilon$, assuming Φ has zero mean and unit variance, and y has zero mean.

Result: The set of chosen atom index $\mathcal{K}$, chosen orthonormalized atoms $\mathbf{D}$, solution vector z w.r.t $\mathbf{D}$, solution vector x w.r.t Φ.

initialize residual $r \leftarrow y$, $\mathcal{K} = \emptyset$, $\mathcal{K}^c = \{1, 2, \ldots, M\}$, $\Psi = \Phi$, $\mathbf{D} = \mathbf{0}$, iteration counter $t = 1$;

while $\|r\|_2 >= \epsilon$ **do**

1. Update remaining atoms $\Psi \leftarrow \Phi(:, i)$, $\forall i \in \mathcal{K}^c$;
2. $\phi_t \leftarrow \text{argmax}_{\phi_n \in \Psi} |\langle \phi_n, r \rangle|$ and $k \leftarrow$ index of ϕ_t;
3. Update chosen index set $\mathcal{K} \leftarrow \mathcal{K} \cup \{k\}$ and remaining index set $\mathcal{K}^c \leftarrow \mathcal{K}^c \backslash \{k\}$;
4. Orthonormalize ϕ_t w.r.t $\mathbf{D}$: $\psi_t \leftarrow \phi_t \perp \mathbf{D}$;
5. Augment $\mathbf{D} \leftarrow [\mathbf{D}|\psi_t]$;
6. $z_t \leftarrow \langle r, \psi_t \rangle$;
7. Update $r \leftarrow r - z_t \psi_t$;
8. Augment $z \leftarrow [z|z_t]$;
9. $t \leftarrow t + 1$;

end

(*Optional*) Denote the chosen original atoms indexed by $\mathcal{K}$ as $\Phi_{\mathcal{K}}$, solve least square problem $x = \Phi_{\mathcal{K}}^{\dagger} y$.

We note that the solution vector z can be written as

$$z = \mathbf{D}^T y. \tag{2}$$

This is because $z_t = \langle r_t, \psi_t \rangle$ in Step 6 can also be written as $z_t = \langle y, \psi_t \rangle$ because y can be represented as linear combination of previously found orthonormalized atoms (which are all orthogonal to ψ_t) and r_t.

To show Algorithm 1 yields identical solution to the original OMP, let us examine the QR decomposition method commonly used for the least squares update. For a residual $r = y - \Phi_{\mathcal{K}} x$, denote the QR decomposition on the chosen atom matrix $\Phi_{\mathcal{K}}$ by $\mathbf{QR}$. The $\mathbf{Q}$ matrix is exactly the $\mathbf{D}$ matrix in Algorithm 1 by the construction of the QR decomposition. To show that the original OMP solution is equivalent to Algorithm 1, we just need to show that $\mathbf{R}x$ is exactly equal to Z from Algorithm 1, when $\mathbf{Q} = \mathbf{D}$. Left multiply $\mathbf{Q}^T$ on the residual equation, we get the following

$$\mathbf{Q}^T r = \mathbf{Q}^T y - (\mathbf{Q}^T \mathbf{Q})\mathbf{R}x$$
$$= \mathbf{Q}^T y - \mathbf{R}x \tag{3}$$

To minimize $\|r\|_2$ (least square), it is equivalent to solve the system of linear equations by setting Eq. 3 to zero.

$$\min \|r\|_2 \Leftrightarrow \mathbf{R}x = \mathbf{Q}^T y \tag{4}$$

The right side of the above equation is identical to the solution in Eq. 2. Therefore, Algorithm 1 is an alternative way to give the same solution as the original OMP.

Our proposed eOMP algorithm is motivated by Algorithm 1. Notice that the update in Step 7 of Algorithm 1 will lead to a residual reduction norm of $|z_t|$ with each new atom. Therefore, to maximize the residual reduction by each new atom, we should maximize $|z_t|$, the magnitude of the correlation of the current residual with the othonormalized chosen atom. However, in Step 2, the atom is chosen based on the correlation with the original atoms. Only after an atom is chosen based on this correlation, then the atom is orthonormalized

(Step 4), and the coefficient is obtained using the projection to the orthonormalized atom (Step 6).

To maximize the error reduction at each new iteration, eOMP algorithm first orthonormalizes all the remaining atoms with respect to all previously chosen atoms, and then compute the correlation of the current residual with these othonormalized atoms and find the one with the largest correlation magnitude. Instead of orthonormalizing each remaining original atom with respect to all previously chosen atoms, we further propose to perform the orthonormalization recursively, i.e., each time a new atom is chosen, all remaining atoms are orthonormalized with respect to this new atom only, and the original remaining atoms will be replaced with the newly othonormalized ones.

More specifically, starting at the second iteration, the algorithm recursively performs a one-step orthonormalization of all remaining atoms with respect to the chosen atoms. Denote the last orthonormalized atom in the chosen set $\mathbf{D}$ by d, the one-step orthonormalization operates on each remaining atom ψ_i by:

$$\psi_i \leftarrow \psi_i - \langle d, \psi_i \rangle d$$
$$\psi_i \leftarrow \frac{\psi_i}{\|\psi_i\|_2} \tag{5}$$

The orthonormalized atoms, instead of their original atoms in the sampling matrix, are then used to evaluate the correlation with the current residual and the one with the highest correlation (in magnitude) is chosen as the next atom. The computation of the new coefficient and new residual can be done using the newly chosen atom, which is already orthonormalized.

The proposed algorithm is presented in Algorithm 2. We highlight the differences of our proposed eOMP from Algorithm 1.

Algorithm 2: Proposed eOMP algorithm with recursive orthonormalization of remaining atoms

Data: $\Phi \in \mathbb{R}^{N \times M}, y \in \mathbb{R}^{N \times 1}, \epsilon$, assuming Φ has zero mean and unit variance, and y has zero mean.

Result: The set of chosen atom index $\mathcal{K}$, chosen orthonormalized atoms $\mathbf{D}$, solution vector z w.r.t $\mathbf{D}$, solution vector x w.r.t Φ.

initialize residual $r \leftarrow y$, $\mathcal{K} = \emptyset$, $\mathcal{K}^c = \{1, 2, \ldots, M\}$, $\Psi = \Phi$, $\mathbf{D} = \mathbf{0}$, iteration counter $t = 1$;

while $\|r\|_2 >= \epsilon$ **do**

1. Update remaining atoms $\Psi \leftarrow \Psi(:, i)$, $\forall i \in \mathcal{K}^c$;
2. Perform one-step orthonormalization Eq. 5 of all columns in Ψ w.r.t ψ_{t-1};
3. $\psi_t \leftarrow \text{argmax}_{\psi_n \in \Psi} |\langle \psi_n, r \rangle|$ and $k \leftarrow$ index of ψ_t;
4. Update chosen index set $\mathcal{K} \leftarrow \mathcal{K} \cup \{k\}$ and remaining index set $\mathcal{K}^c \leftarrow \mathcal{K}^c \backslash \{k\}$;
5. Augment $\mathbf{D} \leftarrow [\mathbf{D}|\psi_t]$;
6. $z_t \leftarrow \langle r, \psi_t \rangle$;
7. Update $r \leftarrow r - z_t \psi_t$;
8. Augment $z \leftarrow [z|z_t]$;
9. $t \leftarrow t + 1$;

end

(*Optional*) Denote the chosen original atoms indexed by $\mathcal{K}$ as $\Phi_{\mathcal{K}}$, solve least square problem $x = \Phi_{\mathcal{K}}^{\dagger} y$.

There are a few implementation notes to accelerate the proposed algorithm. First, the final least squares step is optional, needed only when a solution vector with respect to the original sampling matrix Φ is desired. Computationally this is equivalent to the last iteration solution update in the original OMP. Note that in applications of

sparse representation for signal compression, it is better to use the coefficients associated with orthonormalized atoms to represent a signal, as in [6]. Second, in Step 3 when selecting the best atom for the current iteration, one could also mark the atoms that have near-zero correlations and exclude them from the consideration in future iterations. Last, one could impose additional convergence conditions in the while-loop. One commonly used condition is to compare the norm of r in the most recent iterations and terminate when the norm stops to decrease significantly.

Now we turn to giving a proof on why eOMP works better. For that, we have the following lemma.

Lemma 1. *In each iteration, eOMP maximizes the residual reduction* $\|\Delta\|_2$, *where* $\Delta := r_{t-1} - r_t$.

Proof. Given an orthonormalized atom ψ, the residual reduction Δ from using that atom is given by $\alpha\psi$, where $\alpha = \langle r_{t-1}, \psi \rangle$. The residual norm is $\|\Delta\|_2 = |\alpha|$.

In eOMP, Step 3 selects a particular orthonormalized atom to exactly maximize $|\alpha|$, since $\psi_t = \arg\max_\psi \langle r_{t-1}, \psi \rangle = \alpha$. Therefore eOMP guarantees to maximize the residual reduction. $\square$

III. NUMERICAL EXPERIMENTS

In this section, we show our numerical experiments in both synthetic and real settings for the proposed eOMP algorithm and compare its performance with original OMP. We first show and compare the exact recovery rate of the signal x with known sparsity, and then the recovered sparsity when the estimate of the observation $\hat{y}$ is numerically identical to y, under unknown sparsity conditions.

A. Synthetic simulations

In the first configuration, we consider the exact recovery rate experiment. Consider an i.i.d. Gaussian ensemble $\Phi \in \mathbb{R}^{N \times 2N}$, with $N = 128$, we construct a k-sparse Gaussian signal $x \in \mathbb{R}^N$ and produce the observations $y = \Phi x$. We use our proposed eOMP algorithm and the original OMP algorithm to run k iterations, and compare the exact recovery rate by measuring whether the recovered signal $\hat{x}$ and x are numerically identical. The above simulation is ran for 500 realizations.

Figure 1 compares the exact recovery rate between eOMP and OMP. Proposed eOMP algorithm still outperforms the original OMP and has a slower dropping rate beyond the critical sparsity point (i.e. the point where algorithm starts to fail exactly recovering the signals).

In the second configuration of synthetic signals, we consider the unknown sparsity case. To evaluate, given a sampling matrix Φ and k, we run both the eOMP and the original OMP until the residual norm $\|r\|_2$ reaches machine precision ($\epsilon < 10^{-5}$). We then measure the L0-norm of recovered signals $\hat{x}$, which we term as the recovered sparsity. We consider two types of sampling matrix, the i.i.d. Gaussian ensemble of $\mathbb{R}^{N \times 2N}$ and the overcomplete DCT (ODCT) with $2\times$, $4\times$, and $8\times$ number of signal dimension. Note that the ODCT atoms have higher mutual coherence than the i.i.d. Gaussian ensemble. The above simulation is also ran for 500 realizations for each k.

Figure 2 shows the recovered sparsity by eOMP and OMP for i.i.d. Gaussian ensembles. Our simulations show that eOMP results in lower recovered K than OMP over the entire range of true signal sparsity K examined (1 to 80). However, to show the difference more clearly, we only show the k range in which the proposed eOMP significantly differs from the original OMP. The result shows when $k \in [40, 70]$, the proposed eOMP can achieve up to more than 20% L0-norm reduction than OMP.
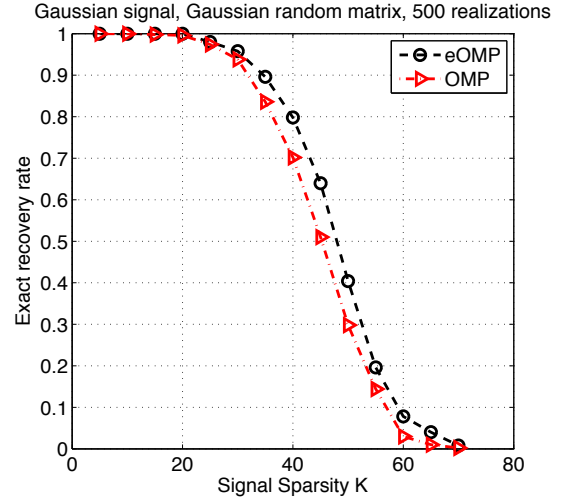

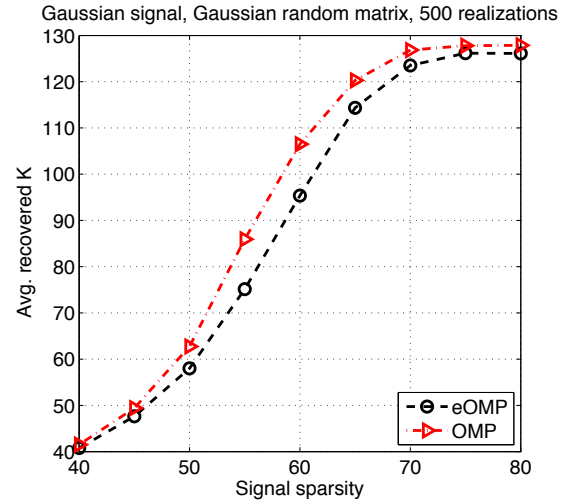Fig. 1. Exact recovery rate for Gaussian sampling matrix

Fig. 2. Recovered sparsity for Gaussian sampling matrix

Figure 3 shows the recovered sparsity of eOMP and OMP for ODCT dictionaries. In all 3 ODCT configurations, we consistently see a large sparsity saving comparing to OMP. The sparsity saving in ODCT cases is larger than the Gaussian ensemble case, which implies our proposed eOMP is more efficient for finding sparse solution when the sampling matrix Φ is more mutually coherent. We note that it is well-knwon that practical audio and image signals can be represented efficiently using DCT and ODCT. Therefore, this is a very encouraging result.

B. Real simulations

In this section, we show a simulation by adopting our proposed eOMP on a realistic application. We examine the gain of the proposed eOMP over the original OMP in the video sparse representation proposed in [6]. The essence of [6] is to find a sparse representation of a video frame block, by using a self-adaptive redundant dictionary consisting of all shifted blocks in the previous frame over a large search range. The motivation for using such a dictionary to represent a block is that the current block is likely to be very close to a few atoms in the dictionary. In the special case when the current block is exactly equal to a shifted block in the previous frame due
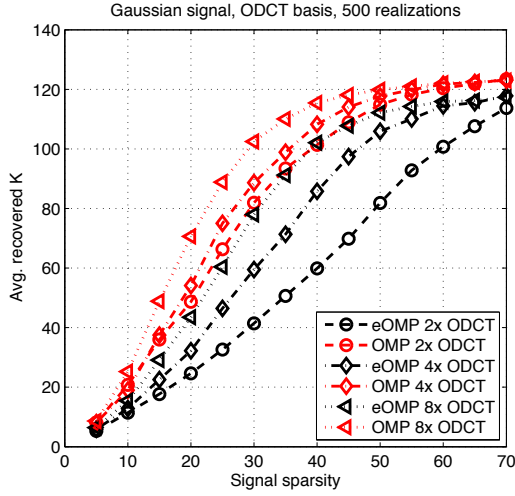
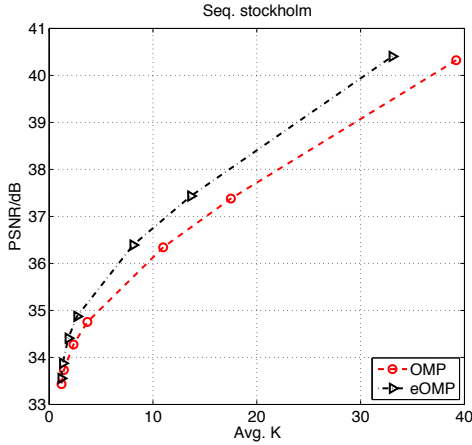Fig. 3. Recovered sparsity for overcomplete DCT basis

Fig. 5. PSNR vs. K for Seq. Football

IV. COMPLEXITY ANALYSIS

The major difference between the original OMP (implemented using Algorithm 1) and eOMP algorithms is the need of recursively performing one-step orthonormalization for all the remaining atoms. Although it sounds tedious, the complexity analysis we will show here implies manageable complexity increase. First, without loss of generality, assume the majority of the complexity coming from multiplications. Therefore an element-wise multiplication of a vector in $\mathbb{R}^N$ is of complexity $\mathcal{O}(N)$. The one-step orthonormalization shown in Eq. 5 includes two inner products and two scaling. To perform a one-step orthonormalization, the total complexity is $\mathcal{O}(4N)$. Note that this has to be done for each remaining atom.

The rest of the ingredients are shared among two algorithms. The correlation calculation between the residual and each dictionary atom takes a complexity of $\mathcal{O}(N)$; the residual/solution update takes a complexity of $\mathcal{O}(2N)$ in each iteration. For OMP using Algorithm 1, orthonormalization one atom with respect to k previous chosen atoms has a complexity of $\mathcal{O}((2k + 2)N)$. Assuming no other numerical trick is used, this complexity is similar to that of solving a least square update by QR decomposition.

Now, consider the case that we run both algorithms on a sampling matrix of $\mathbb{R}^{N \times K}$, with signal in $\mathbb{R}^N$ for s iterations. For original OMP algorithm in Algorithm 1, note that the set of chosen atoms is always increasing, so the orthonormalization process takes a complexity of $\mathcal{O}((1 + \ldots + s - 1)2N + 2sN)$ for s iterations. Therefore the total complexity is:

$$\mathcal{O}((K + K - 1 + \ldots + K - s + 1)N) +$$
$$\mathcal{O}((1 + \ldots + s - 1)2N + 2sN) + \mathcal{O}(2sN)$$
$$= \mathcal{O}\left(\left(\frac{(2K - s + 7)s}{2} + s(s - 1)\right)N\right) \quad (6)$$

For the eOMP, one-step orthonormalization is applied to all remaining atoms, which is decreasing in number, so the second term is replaced by $\mathcal{O}((K - 1 + \ldots + K - s + 1)4N)$ for s iterations. Therefore the total complexity is:

$$\mathcal{O}((K + K - 1 + \ldots + K - s + 1)N) +$$
$$\mathcal{O}((K - 1 + \ldots + K - s + 1)4N) + \mathcal{O}(2sN)$$
$$= \mathcal{O}\left(\left(\frac{(2K - s + 5)s}{2} + 2(2K - s)(s - 1)\right)N\right) \quad (7)$$

Assuming $K \gg s$, Eq. 6 and Eq. 7 are approximately $\mathcal{O}(NKs)$ and $\mathcal{O}(5NKs)$, respectively. The ratio of two complexity terms is

Fig. 4. PSNR vs. K for Seq. Stockholm

to motion, the sparsity is 1. In the example examined here, the block size is 16×16, and the search range of $(-23 \sim 24)$ shift in both horizontal and vertical direction is used, leading to a dictionary of size $N = 16 \times 16, M = 2304$. Furthermore, we do not know the true sparsity of each block.

We show the result as PSNR vs. K plots, where the vertical is the recovery fidelity measured by PSNR, and the horizontal is the average L0-norm over all the blocks in a frame. To generate a curve, we vary the recovery fidelity threshold ϵ to approximate different expected distortions of applying uniform quantization on the coefficients with respect to the orthonormalized atoms. Assuming the bits spent on each non-zero entry is roughly the same, the higher curve would indicate better quality for the same bit rate. Figures 4 and 5 show PSNR vs. K for two video sequences called Stockholm and Football, respectively. By using the proposed algorithm, the recovered K for the same reconstruction fidelity (i.e. same PSNR) reduces by up to 30%, which translates to substantially reduced bit rate. This is impressive improvement as this simulation was ran on real signals (images) with a real dictionary that is highly redundant and correlated.
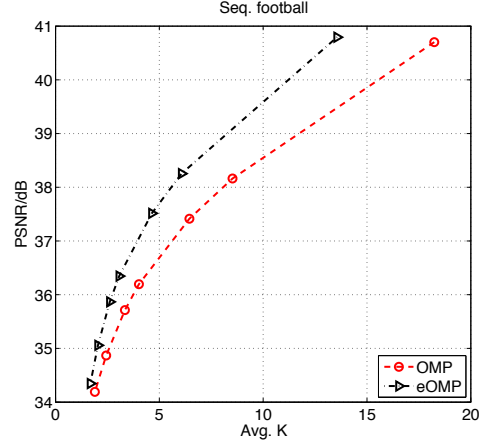
TABLE I
Total running time (in seconds)

Signal sparsity	10	20	30	40	50	60
OMP in SparseLab [7]	0.8614	2.393	4.076	5.506	6.418	6.678
OMP in Algorithm 1	0.6078	1.709	2.889	3.866	4.542	4.666
Proposed eOMP	5.119	11.30	17.45	25.64	33.71	39.79

constantly around 5. To validate this, we measure the total running time for the $2\times$ overcomplete DCT simulation shown in Sec. III-A for both the original OMP and the eOMP. For the original OMP, we use the implementation provided in SparseLab [7] software, which solves the least squares update using QR decomposition. Table I lists the total running time for all three algorithms, implemented in Matlab, for 500 trials of each given K. We see the ratios are around $5\times$. The slightly larger ratio may be due to the inefficient implementation of our eOMP algorithm.

V. Conclusion

In this work, we propose an improved OMP algorithm eOMP, by recursively orthonormalizing the remaining atoms. We have proved the eOMP algorithm maximizes the residual reduction at each iteration. Through extensive numerical simulations with both synthetic and real data, we conclude the proposed eOMP algorithm can consistently achieve better recovered sparsity and higher exact recovery rate than the original OMP. The gain is more significant when the signal's actual sparsity is unknown and the sampling matrix (dictionary) is highly correlated. Our complexity analysis shows that the proposed eOMP algorithm has the same order of magnitude of complexity as the original OMP, roughly equals 5 times more complexity than that of the original OMP when the sparsity is significantly smaller than the dictionary size.

The work reported in this paper is by no means complete: we have yet to provide a mathematical proof on the exact recovery condition, and we are looking to possible further ways to reducing the running time. There are several recent theoretical developments on the greedy L0-norm algorithms but as far as we known, none of them are designed with a clear application perspective. For example, the algorithm needs to know the actual signal sparsity, which is typically unknown in practical application. Our goal is to pursue a better, more application friendly L0-norm minimization algorithm.

References

[1] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[2] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[3] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[4] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2230–2249, 2009.

[5] E. J. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203–4215, 2005.

[6] Y. Xue and Y. Wang, "Video coding using a self-adaptive redundant dictionary consisting of spatial and temporal prediction candidates," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1–6.

[7] D. Donoho, V. Stodden, and Y. Tsaig, "Sparselab. 2007," *See http://sparselab. stanford. edu..(Accessed January 23, 2014.)*.